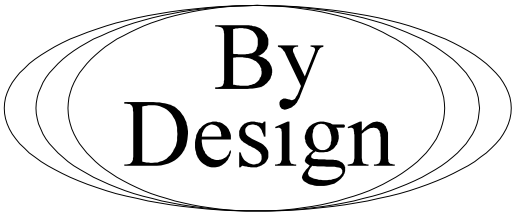

Guidance notes on Native MapInfo Format

A user guide on storage of spatial data in Native MapInfo Tables with a particular focus for users of OSI, OSNI and OSGB large scale data sets such as MasterMap from OSGB

Guide 1 – Bounds Clause

Blackstone – 25 Pendre – Brecon – Powys – LD3 9EA
Tel: (0874) 622569 - Fax (01874) 610677
Vat Number 531 7265 56

By Design

Version 1.0 3 November 2005

CONTENTS

Background

Purpose of this document

Who should read the article

Real World Accuracy

A contradiction ?

Why does the bounds clause exist in MapInfo

Effects of step size

How to specify a bounds clause/step size

Can lost accuracy be restored?

Why might this be important to me

Other factors that can effect accuracy

APPENDICES

APPENDIX A MapInfo Native – the business benefit

APPENDIX B Pros and Cons of Bounds clause

APPENDIX C Pros and cons of Integer versus floating point

APPENDIX D A brief summary of Transverse Mercator projections used in Ireland and Great Britain

APPENDIX E Some commonly used acronyms

APPENDIX F Other possible candidates for "bounds" clause

APPENDIX G MBX - Coordsys Bounds Manager

APPENDIX H About By Design

Background

This article follows some questions on this subject posted to the MUGUKI discussion list, and an answer posted by By Design. It was suggested by Ian Robertson and Graham Walters that By Design submit an article about the Bounds Clause as used by MapInfo for inclusion on the MUGUKI website.

Purpose of this document

In the absence of any other clear guidance, these (this) document(s) recommend(s) a possible way forward for all MapInfo users in Eire and the United Kingdom to store spatial data in native MapInfo tables. This particular document focuses on a recommended bounds clause.

The enclosed covers exactly why the clause is there and how to take best advantage of it.

Who should read the article

The points raised are particularly relevant for users in Ireland (OSNI and OSI) Wales Scotland and England. After the main section we have included some more detailed references in appendices for anyone who wants to delve a bit deeper. There are also a few technical descriptions “up front”, that are necessary to understand this topic. If this first section is not to your liking, read the other sections first, and perhaps return to this technical session, if you then feel the issue is relevant to you.

Real World Accuracy

One recommendation we would make to all organisations using MapInfo is that their users should be trained to understand that they should not expect to zoom in with any CAD or GIS system on OS data and then assume that measurements taken are absolutely correct. The technology allows us to zoom such that real world distances are represented larger on the screen than in real life!

Most datasets are designed to be appropriate for certain scales. For example OSCAR (Centre Alignment of Roads) has been available at three scales, and each one will be best choice for a particular scale and a particular purpose. For representation of National data at say 1:250000 it is actually advantageous to “distort” real world objects such as “Spaghetti Junction” so that users can more easily identify their position.

Even with the largest scale datasets, they are only intended for scales up to approximately 1:500. There is no hard and fast rule, but certainly we should only consider dimensions taken to be correct within half a metre or so.

For example no architect or engineer should use an Ordnance Survey plan to design even a simple garage, without visiting the site, and checking key dimensions. Boundaries of land parcels are in particular often incorrectly represented.

A contradiction ?

This article goes on to recommend defining a bounds clause that holds single millimetre resolution. So taking the above comments into account is this a contradiction of terms of reference?

We believe not. The recommendation for millimetre resolution is more about consistency and precision than about absolute accuracy

Whatever step size is chosen, we recommend you make sure you use the same for all your layers. Better still make sure organisations you share data with use the same step size. A logical progression is for the user group to adopt and recommend a bounds clause for use by all interested parties.

Almost certainly even a resolution of 100 millimetres would be fit for purpose. However as all three mapping agencies considered here deliver data to three decimal places of a metre, the chosen step size might just as well be the millimetre.

If any “CAD” type use of data, such as offsetting, trimming, and particularly extending (extrapolating) lines is to be done then the millimetre will be a better choice, as with each edit errors are likely to accrue.

If any spatial analysis is to be done on different datasets, then they need to be using the same bounds clause. Data can only be “snapped” if it using a bounds clause that is compatible with the data it is being snapped to.

Why does the bounds clause exist in MapInfo

Some of the technical detail here might seem a bit detailed and at first irrelevant, but by going to this depth , it is possible to easily explain exactly why the bounds clause exists in MapInfo.

It might seem a strange place to start but first we must consider the ways that a computer can store numbers. MapInfo users will know that they can choose to store numbers as decimal, ie as text strings, smallint, integer or float within the DAT file (attribute data).

The decimal (text string) is not an efficient way to hold numbers greater than 99. Further, if we wish to manipulate the numbers, ie addition, subtraction etc the computer must first convert the numbers into one of the binary formats and probably back again, if we wish to store a result in decimal. The alternative three binary formats are also more commonly known as:-

MapInfo	**Microsoft**	
Smallint	short	2 byte integer
Integer	long	4 byte integer
Float	double	byte floating point

Also available

-	Single	4 byte floating point
-	BYTE	1 byte integer
-	Int64	8 byte integer

The first two, short and long, are both ways of holding “integer” (or “whole”) numbers – for example 1,2,3,10,101 etc. The short uses 2 bytes per number, the long 4 bytes per number. A double is a way of holding a “floating point” (or “real”) number. Examples 3.14192, 313442.123. A double takes up 8 bytes per number.

So where are we going with this - why is it relevant?

All of the geometry from Ordnance Survey is of course provided as numbers. The Eastings and Nothings are given to three decimal places in metres. This means the smallest unit that they give is one millimetre. Below is an example of the numbers we get eg

E 313244.506 m N 277110.903 m

Initial inspection would suggest the best way to hold these numbers would be floating point ie double. However there is a trade off here. If we could hold them as short or long the file sizes would be a lot smaller, particularly relevant when limited to a 2 GB file size limit.

The MapInfo TAB format started life in 1986. There were still programmes on the TV on “use of” the BBC Micro. The first Amstrad appeared in 1987 with a 5 MB hard disk. MapInfo Professional was introduced as a desktop GIS system and therefore needed to “compress” as much as possible into their files.

The approach MapInfo took was to introduce the concept of a “step” size. Every coordinate pair that is stored in a MapInfo table is stored as two integer number times a step size (or step sizes), which is (are) unique to each

individual table. The step size is set *indirectly* by supplying a bounds clause. If a user does not supply a bounds clause then MapInfo will use a default one, which varies from projection to projection, and thus a default step size.

So the bounds has nothing to do with British National Grid or Irish Grid. It's a method to limit an "area of interest" within a map projection for use within MapInfo. You set the "area of interest" with a bounds clause such as (-5,48) (4, 54). This example would define the area of interest by defining a rectangle 9 wide by 6 high. These values are the difference between the maximum and minimum values for X and Y in the bounds clause which is specified as:

(Xmin, Ymin) (Xmax, Ymax)

The units on the x and y axis are set within the projection and typically are degrees or metres. The example above would be typical for a projection that deploys degrees.

MapInfo divides the rectangle specified into 2000 million steps on the Y axis and 2000 million steps on the X axis. So in the above example the X step would be 9 degrees / 2000 million. The step on the Y axis would be 6 degrees / 2000 million.

The 2000 million (2 GB) is a recurring theme in PC based software. It is half the maximum number that can be held in a 4 byte integer (or long) and half the file size limit imposed by FAT file systems. Integer numbers can be signed or unsigned. Some Microsoft functions limit you to just the positive part of a signed integer, thus the half, to 2 GB. So what MapInfo does is use this range to the full. The minimum bound will be at one extreme of the 4 byte long, the maximum bound will be at the other extreme of a 4 byte long.

I appreciate this seems almost too much information to explain getting best accuracy for OSI, OSNI and OSGB data but now we are in a position to understand exactly what happens to our data using the two different options now available in MapInfo version 8.0.

All three Ordnance Surveys use 1 millimetre as their smallest unit of measurement. Can we store this with absolute accuracy in MapInfo? If we want a step size of 1 millimetre as MapInfo divides the bounding rectangle by 2,000,000,000 we need to set the rectangle to 2,000,000,000 times 1 millimetre. This is a rectangle 2,000,000 metres by 2,000,000 metres (or 2,000 km x 2,000 km). Ireland uses different projection parameters to Great Britain and so the two areas should not be mapped in each others projection. Transverse Mercator has errors that increase on the X axis and each Projection is "designed" for the area to be mapped.

Ireland on its own easily fits in a 2000 km x 2000 km rectangle. Great Britain on its own again fits such a rectangle. In fact Ireland will fit in a 1,000 km x 1,000 km square which would set the internal step size of a table to 0.5

millimetre. However as OSI and OSNI only give to single millimetre there is no advantage in such a small size.

As with most choices in life there are ups and downs with any decision. There is a downside to setting a smaller than necessary rectangle. The smaller the rectangle the bounds clause specifies, the larger the output files will be, and therefore translation can take a little longer. There are also pros and cons to using doubles over longs. Both these subjects are discussed in references at the end for real “anoraks”. Some datasets from Ordnance Survey are rounded to the nearest metre. Also the National Street Gazetteer is rounded to the nearest metre. For users really comfortable with this subject, it is possible to store these layers in a 2,000,000 km x 2,000,000 km bound clause. This will significantly reduce file sizes and still keep full resolution of the data.

Effects of step size

A coordinate can only be stored with a value of a whole number times the step size. For example if the step size was 3.11 millimetre, then values would start at 0, 3.11, 6.22, 9.33 millimetres etc. The Easting 311,000.000 is an exact multiple and could be represented with this step size. The next possible value after this would be 311,000.00311 for a step size of 3.11.

Lets consider a couple of analogies. Imagine the inside storage of the table as a mosaic of tiny tiles. Each tile is a rectangle (could be square) where the length on the x side is the x step and the length on the y side is the y step. Then any coordinate we want to represent can only be stored on one of the corners of any tile. If a point required falls inside a tile, it will get snapped to the closest corner by MapInfo. Setting a bounds clause of 1 millimetre on both axiis sets these snap values to 1 millimetre thus storing Ordnance Survey values correctly. This snap takes place after the user may have “snapped” to a layer with a different bounds clause. So the user may think they have snapped, but the coordinate will not be stored with the same value as the point they snapped to. The point they want is within the area of the tile, but MapInfo stores it on a corner.

The second analogy is currency conversion. Some of us may remember decimalisation of our currency. Four shillings and eleven pence (4s 11d) would equal 25p. However going the other way 25p is five shillings!

How to specify a bounds clause/step size

Whenever you are given a list of possible projections (optionally with bounds clause) the options come from a file called MapInfow.prj. It has always been

possible to add user defined projections and optionally bounds clauses to MapInfo.prj. In version 8.0 of MapInfo Professional the second option of 1 mm precision has been added as standard.

For users of MapInfo 8.0 and later

From options menu choose preferences.

Preferences – Map Window

Map Window Preferences

Category – British Coordinate Systems

British National Grid (1 mm accuracy)

This will then guarantee that on that particular install of MapInfo each new table will have the bounds clause applied to British National Grid which allows storage with absolute precision.

[NB Having set these defaults, you can still check each time a table is created. A “feature” of version 8.0 is that if you do this check, and have set the value to 1 mm, the check will appear that you have the old default (not 1 mm). However this report is wrong - the good news is that the default set in preferences is used].

For users before version 8.0

You need to add a user defined projection to the MapInfo.prj file. Before you do this back up the existing file, in case you make a mistake changing it. Having done this use an editor such as notepad to edit MapInfo.prj

Search for the section headed “--- British Coordinate Systems ---“

Now add a new line to the existing options. The first parameter in quotes is the name that will appear when you try to select a projection. The line should be as below:-

“BNG (1 mm precision)”, 2008, 79, 7, -2, 49, 0.9996012717, 400000, -100000, 0, 0, 2000000, 2000000

(The above should all appear on one line)

Whenever you create a new table, you can now select this new option.

You can also set the default session projection to British National Grid and we would recommend that you do this.

Can lost accuracy be restored?

Unfortunately not. Consider the decimalisation example. Having converted four shillings and eleven pence to 25p, five shillings to 25p and also five shillings and one penny to 25p the original value is then forgotten. If we try and convert default bounds back to 1 mm precision, we will just get the millimetre value closest to the rounded value. So just like all three 25p values above will go to back to five shillings, values in MapInfo will still be rounded in a new bounds clause.

For example consider a bounds clause with a step size of 6.2 millimetres.

0,1,2,3 values from OS would be stored as 0

4,5,6,7,8,9 would be stored as 6.2

10,11,12,13,14,15 would be stored as 12.4

and so on.

Now consider saving these “rounded” values back into a “1 mm precision” bounds clause. All 0s (originally 1,2,3) stay as 0. The original values of 4,5,6,7,8,9 will all now be rounded from a value of 6.2 to 6.0 millimetres. It can thus be seen that some values actually are even less accurate. For example an original 13mm became 12.4 and in the second conversion becomes 12.0.

We appreciate all these values are tiny. However if you are later doing spatial analysis that relies on boundaries being coincident these issues are important. Whilst you might be able to set a tolerance on such calculations, such a “fudge” will now also consider ill fitting boundaries to be in tolerance.

Why might this be important to me

CAD use, sharing data, cleaning data sets, checking for coincidence of line work. For example is this property in the grant area - yes or no.

There is no reason at all why MapInfo native tables cannot be used as the definitive holding of all Ordnance Survey data. When CAD users require Map

extracts these can be cut from MapInfo rather than a second translation from NTF or GZ. CAD users will typically trim and extend lines. A short line that is “rounded “ at each end will when extended take a different and less accurate path to a non rounded line.

Other factors that can affect accuracy

It is also important to check that the scale factor used is the correct one. Whilst errors in this value will not be immediately apparent, if you save data to mid/mif and send to another system like AutoCAD values will be translated incorrectly.

APPENDIX A MapInfo Native – the business benefit

MapInfos native file format is a sophisticated filing system for storing both spatial and non spatial data. The format is often referred to as TAB format. Each “table” should be designed to contain all information on a “theme” such as buildings or roads. The table is defined in four files with an optional fifth file that contains any indexes required on non spatial data (the IND file). The TAB file is an ascii file that defines where the data is held. The data need not be contained in native format. For example the TAB file can define data held in Access, SQL Server, Oracle or DBF as well as Native. When used with raster data, the TAB file is used to identify the raster file, and to register it to a particular projection system. The three other files deployed in native format are DAT which contains non spatial data, MAP which contains geometries, ie spatial data and the ID file which is used to quickly link from DAT to MAP. The MAP file contains style information and an RTREE spatial index, as well as the raw geometry.

There is no cost involved in storing data in Native MapInfo format. Various “translators” or “convertors” are available to store data in the format. The format is reasonably compact and can store large areas of data. The spatial index allows fast retrieval of data.

When a user selects an “area of interest” for rendering to screen or printer, the spatial index insures that only this data , plus a small overhead, is read from the files. This keeps network traffic to a minimum, as well as providing good performance.

Perceived problems with the native format are file size limits, and lack of multi user edits. However for storage of OSI, OSNI and OSGB data these should not be a consideration. Any user should not entertain editing these layers single user never mind multi user. They should be regarded as read only.

For the majority of users the 2GB file size limit is also not a problem. Every county in Great Britain can be represented in native tables without hitting a limit.

If the limit is hit, MapInfo has a feature called “seamless” tables which allow much larger areas to be mapped. National Cover of Great Britain can be held in seamless tables.

MapInfo as already mentioned can also reference data in Oracle, Access, DBF, SQL Server and numerous raster formats. Whilst native MapInfo format is ideal for Ordnance Survey data, these other formats are sometimes required where multi user access is essential.

It is our opinion that storing spatial data in native MapInfo format remains a very viable and cost effective method of deploying corporate GIS throughout Great Britain and Ireland.

The fact that the TAB, ID, IND, MAP and DAT files are highly visible in file explorer implies to many users that the filing system is flat, and that it is all too easy to generate hundreds of files that become difficult to manage.

However MapInfo native format is very capable of being used as part of a relational set of data. MapInfo Professional contains various tools to allow “joins” to be performed and saved.

The management of files and data contained therein, will be a problem for system administrators whatever format is chosen to store spatial and non spatial data.

MapInfo native storage needs no specialist DBA staff, who can be difficult to recruit, to train, and then to retain.

With careful and considered management of spatial data it is the belief of By Design that storage in native TAB format should remain part of any corporate MapInfo users arsenal. We believe there should be a genuine business case before moving individual datasets to other storage formats, each of which will have strengths and weaknesses, be it cost, file limitations, or speed and performance.

APPENDIX B Pros and Cons of Bounds clause

Each object stored has its own MBR, and if its small enough this can be stored in shorts rather than longs. The larger the step size the more likely it is that storage can use short rather than long. So Ordnance Survey data stored in a default bounds clause will take up less space and therefore will generally have been quicker to translate than the (0,0) (2000000,2000000) bounds clause.

This however is easily outweighed by the benefit of storing coordinates precisely. For both Ireland, and for Great Britain a bounds clause of (0,0) (2000000,2000000) will force an internal step size of 1 millimetre for that tables coordinates.

APPENDIX C Pros and cons of Integer versus floating point

The benefit of using integer numbers multiplied by a step size is that file sizes are smaller, disk read and disk writes are faster. Translation can theoretically be faster. These considerations were particularly important in the late 80's when a typical PC had a 5MB or 10 MB drive, and an 8086 or 8088 processor. Even today the benefits are still there. The smaller data needed for each object means it takes longer to hit the 2 GB limit that many PC based software packages suffered from. Microsoft Access and MapInfo TAB files both have a limit of 2 GB. If MapInfo had used double to store geometry, a table would have hit this limit with half the data it can hold in shorts and longs.

The approach is particularly suited to a projection that uses metres. However with degrees it is not quite so good as absolute accuracy (distance on ground) will vary with the distance from the equator.

The downside of integer (and the upside of floating point) is accuracy. There is no need to consider use of a bounds clause. A double number allows 14 significant figures of accuracy. This is particularly important when you edit objects.

AutoCAD and ESRI use doubles. In CAD applications this is really necessary. Each time you trim, extend a line, extrapolate a line of sight etc etc there is potentially a rounding error. If a surveyor takes a line of sight between two points say ten metres apart and extends this line 100 metres, the potential 1 mm error becomes 10 millimetres error. Through the design life of a project such errors accrued would be significant.

With modern computers the additional load of floating point over integer is not really a problem. The Amstrad with a 5MB drive has become a Dell with a 250 GB drive. With National Cover of MasterMap taking under 100 GB as integer and 150 GB as floating point the choice is not as critical now as it was twenty years ago. Translation of either storage can be completed in less than 15 hours.(MapInfo TAB and ESRI SHP). Retrieval of either type of data when deploying an RTREE index is perfectly acceptable.

APPENDIX D A brief summary of Transverse Mercator projections used in Ireland and Great Britain

To “project” the curved surface of the earth onto a flat surface like paper or a computer screen, we need to define a projection system. No representation on paper can truly represent the curved surface. Some will represent areas exactly but not distances, and vice versa.

The larger the surface area you need to map, the worse these problems become. There are many different “Projection systems” ie ways to project. For relatively small areas like Ireland and like Great Britain, the Mercator Projection system is the obvious choice. The projection effectively “wraps” our screen or sheet of paper along one of the axiis. Standard Mercator wraps around the horizontal whereas Transverse Mercator as used in Ireland and Great Britain “wrap” about a line of longitude (the vertical axis). Measurements on this y axis passing through the true origin are represented correctly. However errors on the X axis increase as you move in either direction from this central axis. There are also errors on Y away from the central axis. A “scale factor” is used to “share” this error more accurately across the country. There are two vertical lines either side of the central axis where measured values are correct. Between these lines errors have an opposite sign to errors outside of these lines. This is why you should not view Ireland within British National Grid or view Great Britain within the Irish Transverse Mercator Grid.

[MapInfo deploys a projection for each and every Table. It also deploys a projection for each and every Map Window which can be set by the user. If you wish to view the two areas together you should set the map projection to one suitable for viewing the two together. MapInfo will then on the fly transform all layers to use the Map Window projection] .

To minimise the errors the true origin of the projection will nearly always be horizontally at the centre of the area being mapped. The y value for true origin is not important as all values measured on the vertical axis passing through the origin are exact by definition of the system. If this true origin is deployed for coordinates then users will see negative x values for half of their country. This is the case in Denmark for example. Many countries additionally set a “false origin”. This is defined by specifying a horizontal distance and a vertical distance from the false origin to the true origin. Both the systems discussed here set a false origin.

Further parameters needed to define an explicit projection are units of measurement and the ellipsoid to project from. The ellipsoid is a “spun” ellipse defined by a major and minor axis. To be able to transform from one ellipsoid to another you can further specify shifts in X, Y and Z at the centre of

the earth and even rotations on each of the three axiis again at the centre of the earth.

Below are listed all the parameters to define Irish Transverse Mercator Grid and British National Grid as used by MapInfo. As stated before the bounds clause is not part of the projection system. It is a “feature” of MapInfo that controls accuracy of stored geometric data.

Irish Transverse Mercator Grid

2008,	2000 (for user defined) plus 8 Transverse Mercator
42,	MapInfo 42 Datum using Modified Airey ellipsoid
7,	Unit deployed is the metre
-8,	Longitude at true origin
53.5,	Latitude at true origin
1.000035,	Scale factor for “sharing” error across Ireland
200000,	X distance from true origin to false origin
250000	Y distance from true origin to false origin

Recommend Bounds (0,0) (2000000,2000000)

British National Grid

2008,	2000(for user defined) plus 8 for Transverse Mercator
79,	MapInfo 79 Datum OSGB 36, Airy ellipsoid
7,	Unit deployed is the metre
-2.0,	Longitude at true origin
49.0,	Latitude at true origin
0.9996012717,	Scale factor for “sharing” error Cornwall to Southampton to Dover
400000,	X distance from true origin to false origin
-100000	Y distance from true origin to false origin

Recommend Bounds (0,0) (2000000,2000000)

For more information refer to Appendix H in MapInfo Professional User Guide.

APPENDIX E Some commonly used acronyms

BNG – British National Grid. Described above in Appendix D.

GML Geographic Markup Language. This is a file format for transferring Geographic data sets. It is used by Ordnance Survey for supplying MasterMap (although generally compressed as GZ (see below)). It is widely supported as an “open” format and will almost certainly become more commonplace.

GZ A file format that stores compressed data. Ordnance Survey use GZ to compress MasterMap typically to a twentieth of its original size. Such compression is possible because GML is a “verbose” format – ideal for compression.

MBR – Minimum Bounding Rectangle. Each MapInfo table will have an MBR. The MBR for a table is the smallest orthogonal rectangle that contains all objects within the table. This might well change as you add and remove objects from a table. Each object in the table will also have an MBR. Imagine a vertical line completely west of all parts of an object. Keep moving this line eastward until it just touches the most westerly part of the object. Do this another three times mving a line west until it just touches most easterly point, a horizontal line southward until it touches the most northerly point and finally a horizontal line northward until it touches most southerly point. The rectangle enclosed within these four lines is the objects MBR.

NTF National Transfer Format. Defined in a British Standard (BS 7567) this is a file format designed for transfer of Geographic data. It is used by OSI, OSNI and OSGB for distribution of their datasets.

OSI Ordnance Survey Ireland - Dublin

OSNI Ordnance Survey Northern Ireland

OSGB Ordnance Survey Great Britain - Southampton

APPENDIX F Other possible candidates for “bounds”

The default bounds clause of 8.24 mm resolution on the x axis and 9.997 on Y does not have any particularly good characteristics. Whilst 9.997 might seem close to 10 millimetres, it is not 10 millimetres. Bearing in mind measurements start from the origin, by the time you get to West Wales the multiples of 9.997 could be falling anywhere.

At stated at the start of this article, no user should expect absolute accuracy of 500 millimetres never mind one millimetre.

Provided all users are aware of a chosen “bounds” and the results of taking a particular bounds clause there are four clear choices, for users working in metres such as all Ordnance Survey customers.

There are some advantages in using larger step sizes. More objects will fit within 6500 steps, so more objects are stored in less space, and are retrieved and rendered to screen quicker.

Also, not to be recommended, but possible none the less is to map objects such as North Sea Oil rigs that might otherwise not fit in our 2000 km bounds clause. The British National Grid has been designed to be accurate for Great Britain. However with Transverse Mercator errors will rapidily increase as you move East or West of the coastline (see Appendix D).

The four candidates are

1 metre resolution (specify a 2,000,000 km square for bounds)

0.100 metre resolution (specify a 200,000 km square)

0.010 metre resolution (specify a 20,000 km square)

0.001 metre (specify a 2,000 km square – ie new MapInfo Version 8.0 option)

The “squares” defined do not have to start from (0,0). For example a 20,000 km square might be defined with bounds of (-5000000,-5000000) (15000000,15000000).

The 1 metre step is a good option for data that is already rounded to 1 metre. For example Meridian 2 from Ordnance Survey can be stored with no loss of resolution and draw faster than using MapInfo default or any other value listed here as objects whose MBR is less than 65 km (most if not all) are stored as short integers.

APPENDIX G MBX - Coordsys Bounds Manager

MapInfo have for some time provided a free MBX utility called Coordsys Bounds Manager. We would recommend you avoid this like the plague!

It allows a user to “optimise” the bounds clause. This “optimisation” looks at the maximum and minimum values used in the table to data, and assumes these will stay the maximum and minimum values.

Even if they do stay the max and min values, the step size is unlikely to be exactly divisible into one millimetre. Whilst the step size will likely be small, there will still be a loss of resolution.

Perhaps even worse is consideration of PAI and administrative boundary changes. It is quite likely within a change only update from Ordnance Survey that the Bounding rectangle for the table will become bigger. Any new data following outside the “optimised” bounds will not be visible within the table until the bounds are increased.

Also as already stated the precision is already lost so there is no advantage in making the bounds smaller. In fact there is a disadvantage as the tables will likely be bigger as more use of 4 byte longs, and less use of 2 byte shorts is a likely outcome of a smaller step size.

APPENDIX H About By Design

By Design have been partners of Ordnance Survey (OS) since 1989 and have worked with (OS) digital data for use in AutoCAD and MapInfo since 1989.

By Design became MapInfo partners in 1992. One of our focuses since that time has been writing and supporting translators for NTF, CSV and MasterMap data from OSI, OSNI and OSGB. Speed and accuracy have always been key design goals and the (0,0) (2000000,2000000) bounds clause has always been used.

By Designs customers are primarily in the Public Sector and therefore have access to Ordnance Survey data. Another focus of By Design has been to supply low cost viewers that meet licensing conditions of these valuable datasets and to supply common customer requirements. These include copyright conditions and watermarks, rotation and print to scale.

We also supply solutions for web based mapping and data collection on Pocket PC and MobileMapper CE. We can be found at www.MapsByDesign.co.uk.